



United International University (UIU)
Dept. of Computer Science and Engineering (CSE)
CSE 1111 – Structured Programming Language

Final Exam Question Patterns

Functions

1. There is a magical world of Narnia, where time is different from the time in this world and where animals can speak. The path to Narnia is through a cupboard. A very special cupboard which can also store items.
 - (i) Suppose there are some drawers in the cupboard. Each drawer has different number of items stored. In main() function, declare two arrays, items[] and add[] of size 1000. Take an integer n and n integers to populate both the arrays from user.
 - (ii) Write a function additems(int items[], int add[], int n) which will take the declared arrays and n as the parameters and then increase every ith element of the array items by the corresponding ith element of the latter array. (Hint. If items[1]=10, add[1]=4 updated items[1]=14)
 - (iii) Now to open the door of the cupboard, a special password should be uttered. Write another function openDoor(char password[]) which match the parameter password with the predefined password, "Narnia". If it matches, it will print a line- "Door to Narnia is open.". Otherwise, it will print- "There is no door".
 - (iv) In the main function, (a) after declaration and population of the arrays (as mentioned in (i)), (b) call the function additems passing arguments. (c) Then take a string as a user input and call the openDoor function passing that string as argument.

2. Mr. Tunam is having a very bad semester due to the course named SPL. He is very worried about this course because he didn't obtain good marks in the class tests (CTs). Help him to know the marks of his class tests that will be counted in the final grading so that he can prepare adequately for the final exam. Now, write a C program based on the following requirements:
 - i) Write a function input_CT_marks(float ct_marks[], int n), where ct_marks array stores the marks he has already obtained in all the class tests, n is the number of class tests he has attended, that takes the class test marks from user.
 - ii) Write a function take_Highest_CT(float ct_marks[], int n), which will find and return the highest of the all CT marks.
 - iii) In the main() function, (1) declare and initialize the variables and arrays as needed, (2) call the input_CT_marks and the take_Highest_CT functions, and (3) display the returned value.

3. Write a C program according to the following:
 - i. Write a function int factorial(int n) that will return the factorial of a given number. Factorial of a number can be calculated by multiplying the numbers from 1 to n consecutively. For example, factorial of 4 = 1×2×3×4 = 24. Assume n will not be greater than 10.
 - ii. Write a function int sum(int a, int b) that will return the sum of two given numbers.
 - iii. In the main function, take three integers as inputs and calculate the sum of the factorial of those integers using the above functions factorial() and sum(). Note that you cannot make any modifications to the previously defined functions.

4. Write a C program according to the following:
- Write a function "IsEven" that will take an integer as parameter and determine whether it is even or not. If the integer is even, then the function will return 1. Otherwise, it will return 0.
 - Write a function "ComputeEvenSum" that will take an array of integers and n as parameters. n is the number of values in the array. It computes the sum of the even numbers in that array and returns the sum. You must use "IsEven" function for even checking.
 - In main() function, declare and initialize the array and any other variables as needed. Then call the function "ComputeEvenSum" with appropriate parameters and finally display the returned value.
5. Implement a function updateBalance that takes four parameters: an array of floats representing customer balances, an integer representing the customer's unique identifier (which is the index of the array), an integer representing the type of transaction (1 for withdrawal, 2 for deposit), and a float representing the transaction amount. The function should update the customer's balance based on the transaction type, ensuring that withdrawals do not result in a negative balance.
- In the main function, initialize an array of floats to store the initial balances of 100 customers. Take the initial balances from user as input.
 - Take three values: an integer (customer identifier), an integer (transaction type), and a float (transaction amount) from user.
 - Call the updateBalance function passing these values. If the transaction is a withdrawal and the withdrawal amount is exceeding the available balance, then print "Not sufficient balance" and do not activate the withdrawal.
 - If transaction is successful in step (iii), print the updated balance of the customer

6. Find the output of the following program

```
#include <stdio.h>
int ara[5], x = 20;
void change(int p) {
    --p;
    p--;
}
void update(int n) {
    for(int i = n - 1; i >= 0; i--){
        ara[i] -= x;
        change(x);
    }
}
void main() {
    int n = 5;
    for(int i = 0; i < n; i++){
        ara[i] = (i + 5) * 2;
    }
    update(n);
    for(int i = 0; i < n; i++){
        printf("%d, ", ara[i]);
    }
}
```

7. Find the output of the following program

```
#include<stdio.h>
int a,b,c=5,d=2;
int func1(int a,int b){
    c=a+b;
    return c*2;
}
int func2(int p){
    p=p+a;
    a*=2;
    return p;
}
void func3(int d){
    int c=d+10;
    b=func2(c);
}
void main(){
    a=1;b=3;
    printf("%d %d %d %d\n",a,b,c,d);
    a=func1(a,b);
    printf("%d %d %d %d\n",a,b,c,d);
    d=func2(c);
    printf("%d %d %d %d\n",a,b,c,d);
    func3(b);
    printf("%d %d %d %d\n",a,b,c,d);
}
```

8. Find the output of the following program

```
#include<stdio.h>
int x = 2, y = 3;
int fun1(int n){
    return n%11;
}
void fun2(int arr[], int n){
    for(int i = 0; i<n; i++){
        x = fun1(x) + fun1(y);
        arr[i] = arr[i] + x;
        y = fun1(y) + fun1(x);
    }
}
int main(){
    int a[] = {2, 3, 5, 7, 11};
    fun2(a, 5);
    for(int i = 0; i<5; i++)
        printf("%d ", a[i]);
}
```

9. Find the output of the following program

```
#include<stdio.h>
int x = 2, y = 3;
int fun1(int n){
    return n%11;
}
void fun2(int arr[], int n){
    for(int i = 0; i<n; i++){
        x = fun1(x) + fun1(y);
        arr[i] = arr[i] + x;
        y = fun1(y) + fun1(x);
    }
}
int main(){
    int a[] = {2, 3, 5, 7, 11};
    fun2(a, 5);
    for(int i = 0; i<5; i++)
        printf("%d ", a[i]);
}
```

10. Find the output of the following program

```
#include<stdio.h>
int a=0,b=0, c=0;
int func1(int p) {
    c=p+a;
    return c;
}
int func3(int c){
    c = 2;
    a *=2;
    return c*a;
}
void func2(int x, int b){
    x *= 2;
    b = func3(x);
}
void main(){
    a = 2121 % 47;
    func3(a);
    printf("%d %d %d \n",a,b,c);
    b = func1(a);
    printf("%d %d %d \n",a,b,c);
    func2(a,b);
    printf("%d %d %d \n",a,b,c);
}
```